TRANSPLINE CURVE REPRESENTATION SYSTEM

BY

Brian A. Barsky and Spencer W. Thomas

UUCS-80-104

April 1980

TRANSPLINE CURVE REPRESENTATION SYSTEM

by

Brian A. Barsky and Spencer W. Thomas

Department of Computer Science
University of Utah
Salt Lake City, Utah 84112
U.S.A.

This work was supported in part by the National Science Foundation under grant number MCS-78-01966.

Abstract

An interactive curve representation system has been developed based on the concept of transforming among several parametric spline curve formulations. The available formulations are the interpolatory spline, uniform B-spline, spline under tension, and NU-spline. The system implementation is described in the context of a sample design session.

Table of Contents

List of Figures

I. Introduction

Various mathematical techniques of curve representation have been developed and each has properties that make it more appropriate in certain situations than in others. Thus, it is very beneficial to have the capability of transforming between formulations so as to be able to exploit the particular advantages of one formulation when they are most needed [1].

The TRANSPLINE interactive curve representation system demonstrates the application of this concept to a set of four parametric spline curve formulations, each of which was selected on the basis of its appropriateness to a different phase of design. The interpolatory spline is ideal for initial data entry since it is much easier to establish a rough approximation of a desired curve shape by specifying points which lie on the curve, rather than points which lie near the curve. The uniform B-spline is well suited to the modification of already-entered data because the local nature of a B-spline allows a change to be made to one portion of the curve without altering the remainder of the curve, as would be the case if an interpolatory spline formulation were modified. The spline under tension and NU-spline are appropriate for the final shape definition because of the capability of the former to flatten spline segments which may have too great a curvature, and of the latter to sharpen corners while maintaining the desired continuity properties of the spline.

These four types of spline curve formulations have been implemented by the authors in a single system utilizing an Evans and Sutherland Picture System. The initial implementation was under the RT-11 operating system on a DEC PDP-11/05 minicomputer using a Picture System I. The program has since been transported to a DEC PDP-11/34, and a Norsk Data NORD-10 under the SINTRAN III operating system using a Picture System II. It is currently in the process of being implemented under the UNIX operating system.

The implementation provides for the interactive entry and modification of the four types of spline curves, as well as for the transformations among them, as explained in [3]. A detailed description of the system implementation is presented in the following sections.

II. General system overview

The system consists of two major divisions: spline type selection and transformation, and spline entry and modification. The user interacts with these through menus displayed on the Picture System (Figures 1 and 2). The menu is displayed at the bottom of the screen and consists of up to 12 boxes, each with a 16 character label. A menu button may be selected or "picked" by positioning the cursor within the box and depressing the tablet pen. Certain buttons (for example, "quit") cause an immediate action to occur; while others (for example "add spline") merely determine a mode, in which case the desired action is subsequently initiated by depressing the pen with the cursor outside the menu area (within the picture area). A "mode line" is displayed at the top of the screen at all times, informing the user of the exact state of the system. The first (leftmost) field of the mode line contains the name of the type of spline currently selected, the second (center) field displays the resolution value, and the third (rightmost) field always describes the action which would occur if the tablet pen were depressed with the cursor outside the menu area.

The <u>spline type selection</u> menu allows the user to select, via the lower row of menu buttons, the type of spline to be displayed and manipulated. Previously entered splines will be transformed to the selected type automatically when a new type of spline is selected. This is performed with as little loss of information as possible. For example, when transforming from a taut spline (spline under tension or NU-spline) to a non-taut spline, tension information is ignored, although it will be regained if a taut spline type is again selected. The button labeled "resolution" is used to set the number of line segments to be drawn for each spline segment. The resolution value is set by a "graphical potentiometer" utilizing the tablet pen, with an allowed range of 1-9. The button labeled "single tension" is a toggle switch which, when selected, alters the button label to "multiple tension" and allows taut splines to have a different tension value for each segment (splines under tension) or point (NU-splines). If selected a second time, the label will revert to "single tension" and each taut spline will be forced to have a single tension value over the entire curve.

The various buttons on the <u>main spline modification</u> menu perform the following functions: enter splines; add new points to existing splines; modify splines by moving points, vertices, or endpoint first derivative vectors, or by changing the tension value(s); delete splines; delete points from splines; write spline values to a file, or read them from a file; or display any of the polynomial splines as smooth curves by using a subdivision algorithm. Several menu buttons set a mode which will cause the desired action when the pen is depressed in the picture area. Most actions apply only to a single point; those functions which enter a new point ("add spline", "add point") do so at the position of the pen, and those functions which modify an existing point ("move point", "delete point", "delete spline", "change tension") change the point nearest the position of the pen. These operations will be more fully described in the context of a sample design session which will now be presented.

III. Sample Design Session

This section presents a sample design session using the TRANSPLINE system. The sequence followed here is as follows: initial curve entry using interpolatory splines, curve modification to more closely approximate the desired shape using B-splines, and final shape definition using splines under tension and NU-splines.

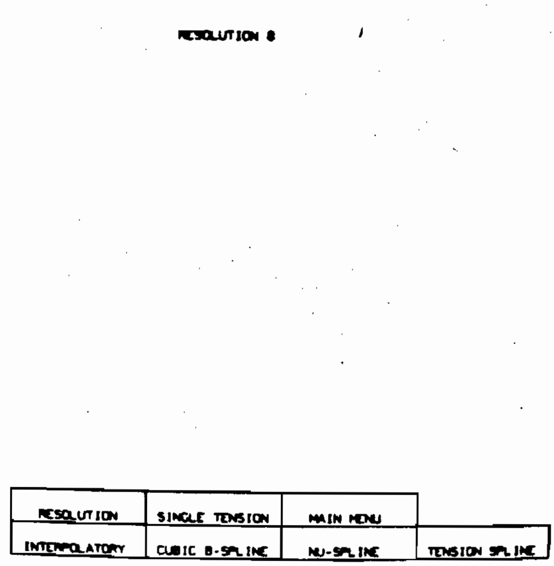

Figure 1.
The initial type selection menu.

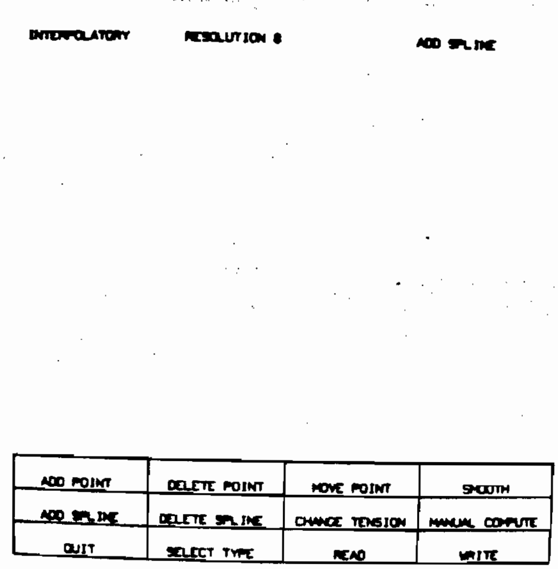

Figure 2.
After picking the Add Spline button.

Upon initiating the spline system, the user is presented with the <u>spline type selection</u> menu (Figure 1). Since no type of spline is initially selected, as indicated by the blank spline type area of the mode line, this is usually the first action taken by the user, who in this case selects the "interpolatory" menu button. After this selection, the user is able to proceed to the <u>main spline entry and</u>

modification menu by picking the "main menu" button. If no spline type had been selected, this button would have no effect, since a spline cannot be entered until its type is defined.

Although any of the buttons on the main menu could now be selected, the only one with a reasonable effect is the "add spline" button (or the "read" button, which will be discussed on page 13). After selecting "add spline" (Figure 2), the user can enter points on the spline by placing the cursor at the desired location and depressing the pen. Each time this is done, an "x" appears marking the position of the point (Figure 3).

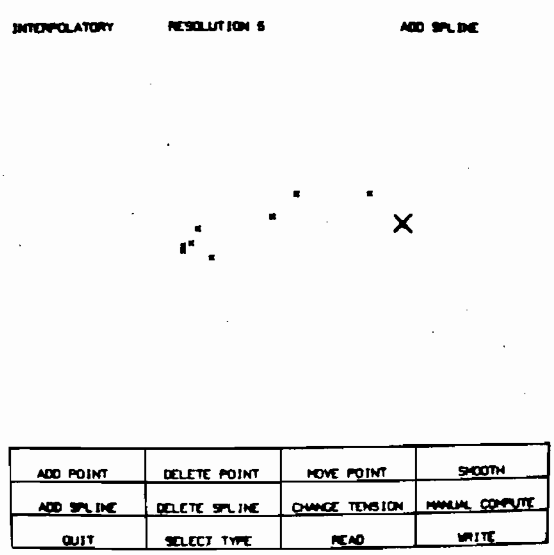

Figure 3.
After entering a few points.

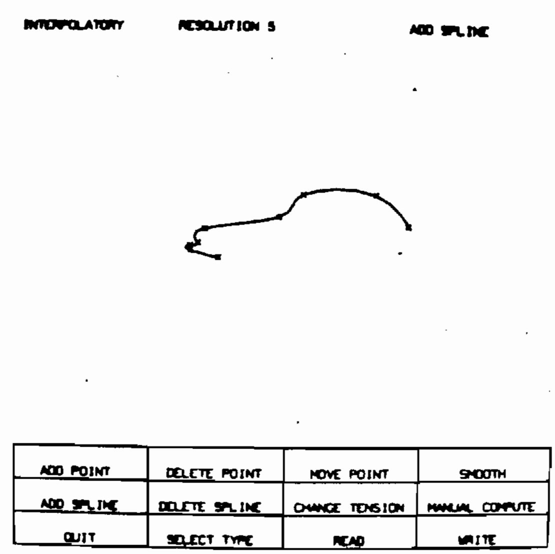

Figure 4.
A computed spline.

The curve can now be displayed by picking the "manual compute" button, causing the system to compute the interpolatory spline passing through the points which have been entered (Figure 4). Note that the action field of the mode line still contains the words "add spline"; thus, more points on the same spline can be entered without picking another menu item. Picking the "add spline" button a second time initiates the input of a second spline which is entered and computed in exactly the same manner as the first spline.

In order to avoid continually selecting the "manual compute" button to see the effect of each change to the splines, the system can be put into "auto compute" mode by picking this button twice in a row, changing the label to "auto compute" (Figure 5). Any subsequent changes to the curves will be immediately reflected in the display by automatically recomputing the splines after each action which modifies one of them. This is generally the preferred mode of operation, but the "manual compute" option is offered for cases where the recomputation time becomes unreasonably long. This choice of modes is unnecessary for B-splines which are always in auto compute mode, since they can be recomputed rapidly.

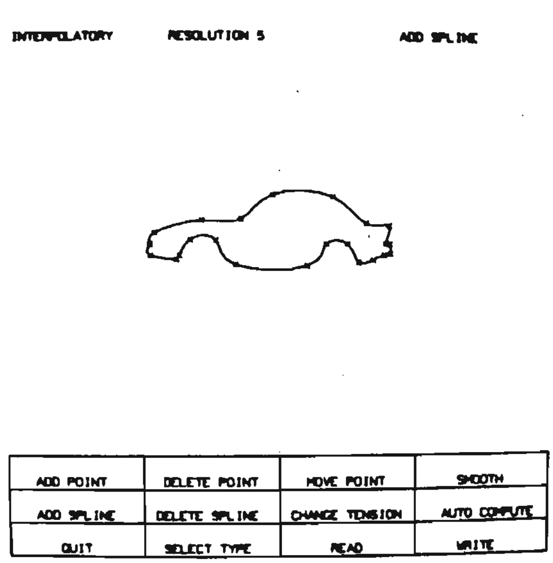

Figure 5.
Setting auto compute mode.

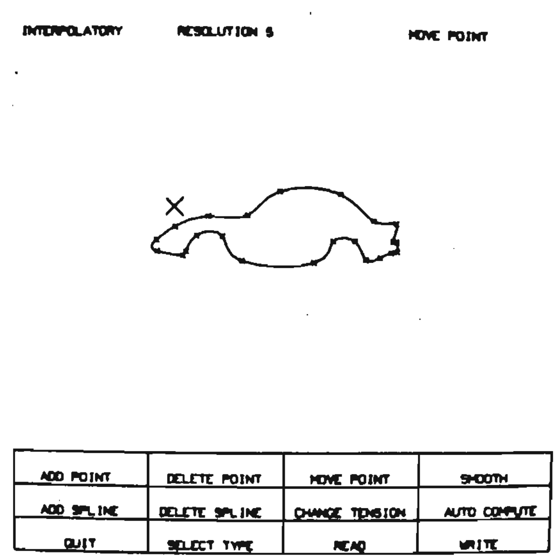

Figure 6.
The spline after moving some points.

After entering this spline, it is observed that it does not yet offer a sufficiently close approximation to the desired shape; thus, "move point" is selected in order to modify it (Figure 6). The "move point" function can be used to modify the position of an interpolated point or B-spline control vertex, or the value of the first derivative vector at either endpoint. The point or vertex which is to be moved is picked by depressing the pen near it, and it then tracks the motion of the cursor until the pen is depressed again. The action is thus that of picking up the point, moving it, and setting it down again. A first derivative vector handle is picked by depressing the pen near either of its endpoints, and it is then moved in the same manner as was described for points.

To transform the spline to a B-spline, the type selection submenu is invoked by the "select type" menu button, followed by the selection of the "cubic B-spline" menu button. This change is immediately reflected in the display (Figure 7). The system is then returned to the main menu by the selection of the "main menu" button.

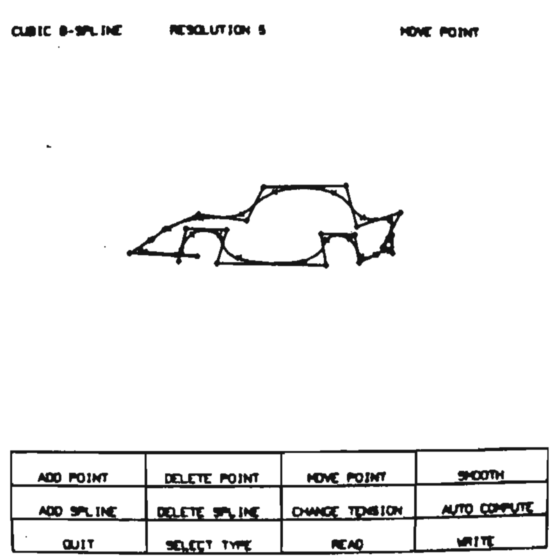

Figure 7.
The spline as a B-spline.

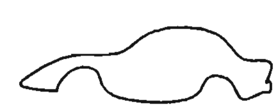

Figure 8.
A smoothly displayed B-spline curve.

The system is still in "move point" mode and the shape modification procedure is continued by moving the B-spline control vertices. This process is facilitated by two properties of the B-spline formulation. First, the local control inherent in B-splines restricts the effects of modifying the position of a single control vertex to a small, predetermined portion of the curve. Secondly, the B-splines are dynamically modified as the control vertex is moved, whereas all the other types of splines would not be recomputed until at least the conclusion of the move operation. B-spline modification in realtime is achieved by exploiting several properties of the B-spline formulation so as to avoid a complete recomputation of the curve. The new curve resulting from the movement of a single control vertex can be computed using a "perturbation" algorithm designed in [2].

At this point, the spline can be displayed as a smooth curve to establish a better perception of its exact shape. The "smooth" button invokes an alternate display method for any of the polynomial splines using a recursive subdivision algorithm (Figure 8). The subdivision is performed by the Lane and Riesenfeld B-spline subdivision algorithm [4] applied to each spline segment. An interpolatory spline can also be subdivided by first transforming to the B-spline formulation, which is done automatically.

Deciding that this is a satisfactory approximation of the shape, the user picks the "write" button in order to save the splines in a file. The "write" button invokes the <u>write</u> submenu (Figure 9), which can be used to write the spline description to a file. The action portion of the "mode line" contains the word "write" and the name of the file to

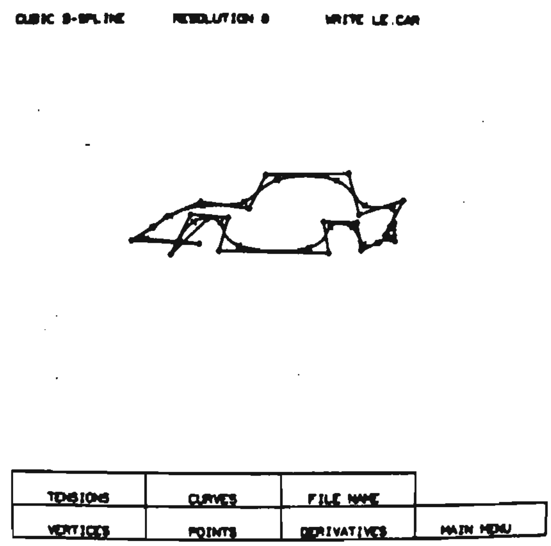

Figure 9. The write submenu.

which the spline data will be written. The "file name" button can be used to change the default file name to any legal file name (for example, the terminal) and prompts for the desired file name on the terminal. The file format is described in Figure 10. The buttons labeled "points", "derivatives", "vertices", "tensions", and "curve" cause the interpolated points, the first derivative vector values (or second derivative vector values in the case of splines under tension), the B-spline vertices, the tension values, or the computed curve, respectively, to be written to the output file. Of course, the stored information will be of little value unless a certain minimum amount of information is written into the file. The necessary information depends on the spline type: for interpolatory splines, the points and derivatives must be written, while B-splines require only vertices, and the taut splines require points, derivatives and tensions to be completely specified. Although it is never necessary to write out the computed curve, this may be done if desired. Finally, the "quit" button closes the output file and returns to the <u>main</u> menu.

Record 1	Column 2	Number indicating spline type (1-4).
	Column 3-20	Text name of spline type, for informational purposes only.
	Column 21-22	Number of splines (currently 1-5).
	Column 23-end	Number of segments per spline, for each defined spline, in nI3 format.
Record 2	Text indicating what type of data follows starting in column 2. May assume values from the set {points, vertices, derivatives, tensions, curve}. Only the first letter is significant.	
Record 3-N	The data indicated by record 2. The format is: for points, derivatives, vertices, and tensions, a separate record for each datum, where an (x,y) pair counts as a single datum; for curves, the first record contains resolution value, and a separate record is written for each spline segment with the points written in n(I6,I5) format. There is a "blank" line at the beginning of the data for each individual spline, which contains a line of text indicating the beginning of the spline and the spline number. This is intended to aid the location of individual data if a human looks at the file. The combination of records 2-N may be repeated as many times as desired. Note that all records have a space in column 1, so FORTRAN prints them on the terminal correctly.	

Figure 10. The format of the file written by the write command.

Having almost obtained the desired shape, the user will now complete the shape definition by using one of the taut spline types. The user thus returns to the type selection menu and picks the "tension spline" button. Since tension does not need to be changed over the entire curve, the "multiple tension" mode is set.

The "change tension" function allows the user to change the tension values for either taut spline formulation. The user picks the segment to be modified, and it is then illuminated while a scale with a sliding pointer is displayed at the bottom of the screen (Figure 11). The system sets the pointer to the current value of the tension, and it can then be moved by positioning the cursor within the "V" of the pointer. When the pen is depressed, the tension value is set and the menu reappears (Figure 12). If the system had been in single tension mode, the entire spline would have been picked, rather than only a single segment. Although the numerical ranges of the tensions are quite different for the two taut spline formulations, the effects of the maximum values of each are quite similar.

Since each tension value in a spline under tension is applied over a spline segment, increasing a tension value tends to "flatten" the segment. This differs from a NU-spline, for which an increased tension has the effect of "sharpening" a corner. In order to start with a "clean slate" with all the tension values set back to zero, the saved B-spline formulation is read back into the system. The user is prompted for the file name at the terminal, and the file is assumed to be in the format created by the write routine (Figure 10).

Then the spline type is changed to "NU-spline" by invoking the type

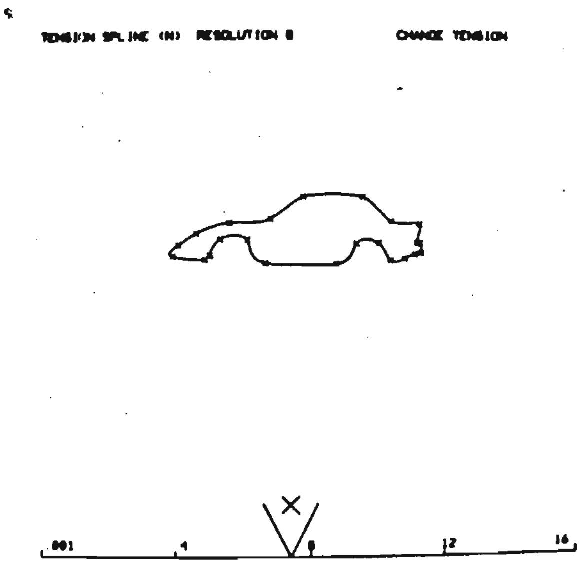

Figure 11.
Changing the tension.

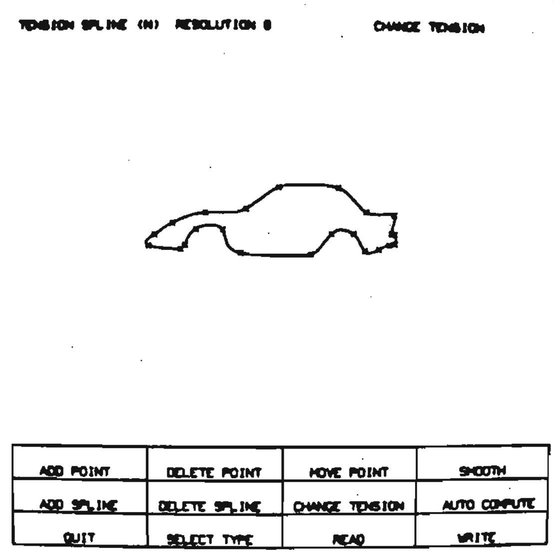

Figure 12.
The resulting spline.

Changing the tension of a spline under tension:
straightening the front of the rear wheel well.

<u>selection</u> submenu and selecting "NU-spline" (Figure 13). The "change resolution" button on the <u>type</u> <u>selection</u> menu can be used to change the number of line segments used to display the curve, whether a small number of segments for faster display update, or a larger number for a more accurate display. Upon selecting the "change resolution" menu button, the resolution is changed by moving the cursor from left to right across the screen, until the desired resolution value is shown in the mode line (Figure 14). The system is then returned to the <u>main</u> menu by the selection of the "main menu" button.

Since the tension values for the NU-spline are associated with the interpolated points rather than the spline segments, it is necessary to pick the point at which the tension is to be changed. This is done by placing the cursor on the desired point and depressing the pen (Figures 15 and 16). The selected point is illuminated for identification while

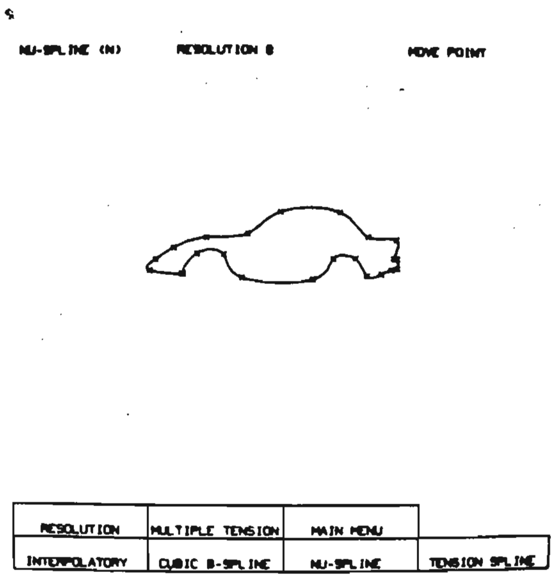

Figure 13.
Changing to a NU-spline.

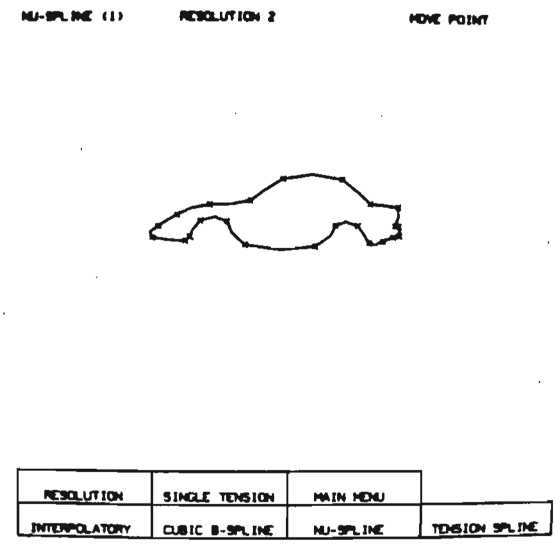

Figure 14.
Changing the resolution value.

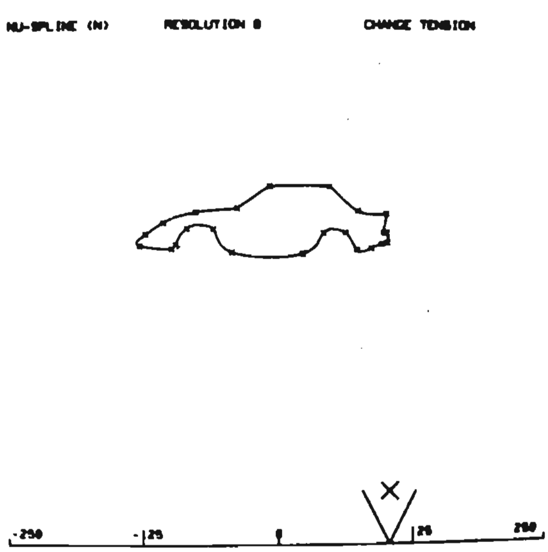

Figure 15.
Changing the tension.

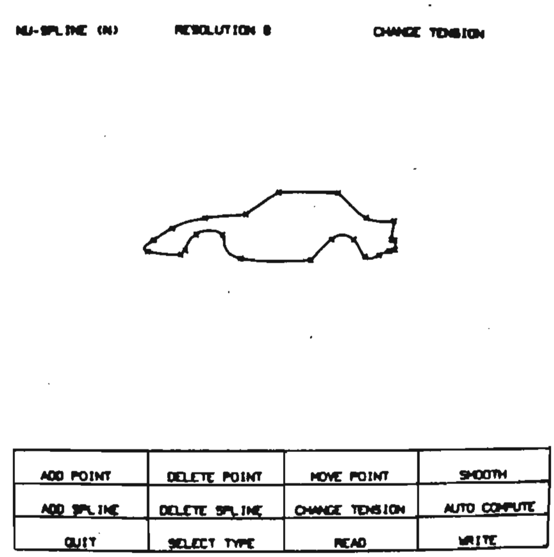

Figure 16.
The result of the change.

Changing the tension of a NU-spline:
sharpening the lower front corner of the rear wheel well.

the tension modification is in progress.

A NU-spline is allowed to have negative tension values, which can generate some interesting curves (Figure 17). Again, the curve can be displayed smoothly smoothly using the "smooth" button. In general, a NU-spline does not have the second derivative continuity which is required for transformation to a B-spline for the subdivision process. However, it can still be displayed by this technique because the subdivision is applied to one segment at a time, and each NU-spline segment can be expressed as a B-spline because it is a cubic polynomial (Figure 18).

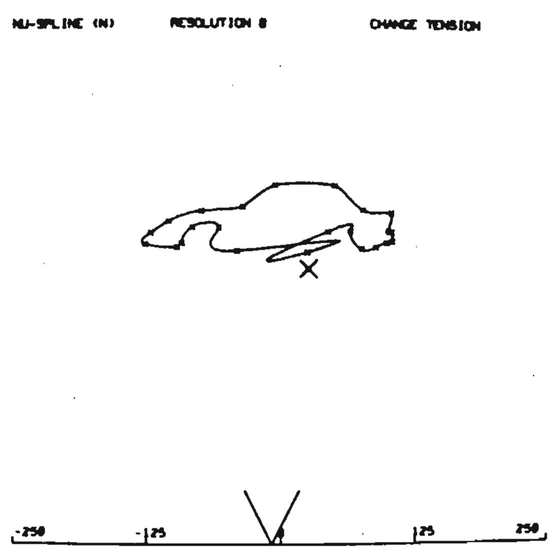

Figure 17.
Negative tension on a NU-spline.

Figure 18.
Smoothly displayed NU-spline.

The system has the capability of displaying and modifying several splines at once; thus, it can be used to form images of greater complexity than would be possible with a single spline. In Figure 19 several splines have been added to the original automobile profile to create the final image.

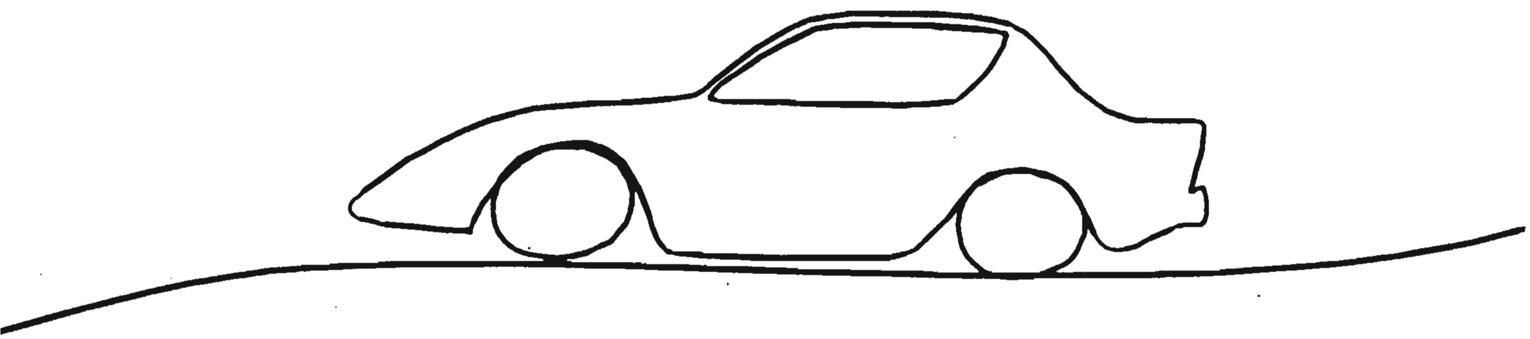

Figure 19. Final automobile profile consisting of several splines.

IV. Conclusion

Each of the various techniques of curve representation which have been developed in the field of computer aided geometric design has properties which are more useful in certain phases of the design process than in others. It is therefore very useful to be able to transform, with as little loss of shape information as possible, between the different formulations in order to select the one most appropriate to the situation at hand.

The TRANSPLINE system is an interactive curve representation system which implements this concept with a set of four parametric spline curve formulations. Each member of this set was selected on the basis of its appropriateness to different design phases; thus, some are better suited to initial data entry, while others are ideal for modification of already entered data, and others are able to supply a particular curve property which the remaining formulations lack. The result is a system which is quite versatile and capable of representing complex shapes, while remaining fairly simple and easy to use.

REFERENCES

1. Barsky, Brian A. A Method for Describing Curved Surfaces by Transforming between Interpolatory Spline and B-spline Representations, Master's Thesis, Cornell University, Ithaca, N.Y., January 1979.

2. Barsky, Brian A. "A Study of the Parametric Uniform B-spline Curve and Surface Representations", in preparation.

3. Barsky, Brian A. and Thomas, Spencer W. "TRANSPLINE -- A System for Representing Curves Using Transformations among Four Spline Formulations", in preparation.

4. Lane, Jeffrey M. and Riesenfeld, Richard F. "A Theoretical Development for the Computer Generation of Piecewise Polynomial Surfaces", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 1, January 1980, pp. 35-46.